

ベクトル空間法を基礎としたカテゴリーマッピング法による Web ページの自動分類

西城 英之^{†1} 鈴木 智充^{†2} 山田 智子^{†2} 富樫 敦^{†3}

^{†1} (有) シルフィード ^{†2} 宮城大学大学院 事業構想学研究所

^{†3} 宮城大学 事業構想学部デザイン情報学科

1 はじめに

Web 上には非常に多くのサイトが存在し、情報で溢れている。そのため、必要な情報を検索するための手段が必要となる。そこで利用されるようになったのが検索エンジンである。検索エンジンには大きく分けて2つのタイプの検索エンジンがある。1つはディレクトリ型検索エンジンであり、もう1つはロボット型検索エンジンである。

Web サイトは通常2つ以上の複数の Web ページから構成されている。Web サイトを訪れる際は、その Web サイトのトップページに遷移する場合とトップページではなく、Web サイト内のユーザの要求を満たす Web ページに直接遷移する場合がある。

通常、トップページには Web サイト内のコンテンツへのリンクが多く、その Web サイトが提供したい詳細な情報やコンテンツは含まれない。そのため、Web サイトのトップページに遷移してきた場合には、ユーザが必要としている Web ページにたどりつくためには、もう1ステップ以上のページ遷移が必要となる。適切なサイトマップがある Web サイトの場合は良いが、適切なサイトマップがない Web サイトの場合は、必要な情報を探るのが困難である。

ユーザの要求を満たす Web ページに直接遷移してきた場合は、その Web ページだけを見て満足してしまい、他の Web ページにもユーザの要求を満たす Web ページが用意されているのに、逃してしまうことが考えられる。

Web サイトの構成はそれぞれ異なるので、Web サイト内のどこにどのようなコンテンツが存在するかを簡単に把握することができるか否かは、Web サイトの作成者に因るところが大きい。また、Web サイトの規模が大きくなればなるほど、コンテンツも増えるのでナビゲーションの善し悪しが重要になってくる。

以上のことから、現在の Web サイトの問題点としては、ユーザが必要なコンテンツが Web サイト内のどこに存在するか把握しづらいということと、必要な情報を逃してしまうということが起こりうることである。

前述した通り、急速な発展を続ける Web であるが、発展の裏には様々な問題を抱えている。本研究ではそれらの問題を解決するため、ディレクトリ型検索エンジンとロボット型検索エンジンの利点を活かした新たな情報検索基盤の提案を行う。それを Web サイトに適用すること

で、Web サイトの問題点を解決することが目的である。

本研究の具体的な手法としては、まず Web ページを分類するための分類体系を構築する。そして、Web サイトに含まれる Web ページを自動的に構築した分類体系にマッピングさせ、Web ページと分類体系とのマッピング情報を検索の基盤として利用する。その過程にある分類体系の構築手法および Web ページを分類体系にマッピングするマッピングアルゴリズムの提案を行う。

提案する手法を用いて、実際に使えるシステムとしての応用可能性についても検討する。応用事例として、意味に基づいた Web サイトマップの自動構築と意味に基づいた Web ページ検索を提案する。

2 Web ページの自動分類

2.1 カテゴリーの定義

Web ページを分類するには何らかの分類体系が必要となる。本研究で利用する分類体系は、ディレクトリ型検索エンジンで利用されているウェブディレクトリと同じ形式をとり、カテゴリと呼ぶ。カテゴリは木構造になっており、階層が深くなればなるほど詳細化されていく。以下にカテゴリの定義を示す。

【定義】(カテゴリ)

カテゴリはグラフ理論の木 $C = \langle C, E \rangle$ として定義される。ここで、 $C = \{c_1, c_2, c_3, \dots, c_n\}$ はカテゴリ名の有限集合、 E は有向辺の関係を与える。 $c, c' \in C$ に対して、 cEc' が成り立つとき、 c は c' の親、あるいは c' は c の子と呼ぶ。また、親は必ず2つ以上の子を持つこととする。

2.2 キーワードの抽出と重み付け

2.1 節で定義されたカテゴリに関連するキーワードを抽出する。抽出されたキーワード群に対して、カテゴリとの関連度に応じた重みを0以上の実数で設定する。カテゴリとキーワード、重みの関係を表1に示す。

【定義】(キーワード、重み、カテゴリの特徴ベクトル)

- (1) $K = \{k_1, k_2, k_3, \dots, k_n\}$ をキーワードの集合とする。
- (2) $W: C \times K \Rightarrow [0, \infty]$ は、カテゴリに対するキーワードの重みを与える関数であり、その値はカテゴリ c に対してキーワード k の帰属性の度合い $W(c, k)$, W

$(c, k) \geq 0$ を表す。ここで $[0, \infty]$ は 0 以上の実数全体の集合を表す。

(3) $c \in C$, $k_1, k_2, k_3, \dots, k_n \in K$ のとき,

$CV(c) = (W(c, k_1), W(c, k_2), W(c, k_3), \dots, W(c, k_n))$ は、カテゴリ c に対する全てのキーワードの重みを表したものであり、 $CV(c)$ をカテゴリ c の特徴ベクトルと呼ぶ。

2.3 重みの自動計算

2.3.1 カテゴリとキーワード更新における問題点

Web ページは世界中に無数に存在する。それらを適当なカテゴリにマッピングするためには、膨大な数のカテゴリやキーワードが必要になる。また、人間が扱う自然言語は文化や技術の発展に伴い、日々変化していく。その変化に対応していくために、カテゴリやキーワードの追加や更新も必要になる。追加、更新処理の計算量は、カテゴリやキーワードの数に応じて飛躍的に増えていく傾向にあるため、計算量を減らすための仕組みが必要と

なる。そこで本研究では、カテゴリの木構造において、下位のカテゴリが存在しない最下位のカテゴリの重みの自動計算方法と下位のカテゴリが存在する上位カテゴリの重みの自動計算方法を提案する。以上 2 つの重みの自動計算方法を適用することで、全てのカテゴリの重みを効率よく自動的に計算することができる。

2.3.2 最下位カテゴリの重みの自動計算

最下位カテゴリの重みを自動計算する方法は、全ての最下位カテゴリに対して、各カテゴリにマッピングされるべき Web ページを複数選択し、それらの Web ページから文章を抽出する。抽出した文章はカテゴリ毎に 1 つの文章として集約する。得られた文章群に対して、文章中の特徴的な単語を抽出するアルゴリズムである tf-idf(Term Frequency-Inverse Document Frequency)を応用した、以下の計算式を用いて重みを計算する。

表 1. カテゴリとキーワード、重みの関係

	カテゴリ c_1	カテゴリ c_2	カテゴリ c_3	...	カテゴリ c_n
キーワード k_1	重み $W(c_1, k_1)$	重み $W(c_2, k_1)$	重み $W(c_3, k_1)$		重み $W(c_n, k_1)$
キーワード k_2	重み $W(c_1, k_2)$	重み $W(c_2, k_2)$	重み $W(c_3, k_2)$		重み $W(c_n, k_2)$
キーワード k_3	重み $W(c_1, k_3)$	重み $W(c_2, k_3)$	重み $W(c_3, k_3)$		重み $W(c_n, k_3)$
...					
キーワード k_m	重み $W(c_1, k_m)$	重み $W(c_2, k_m)$	重み $W(c_3, k_m)$		重み $W(c_n, k_m)$

$$tf_i = \frac{n_i}{\sqrt{\sum_k n_k^2}}$$

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

$$tfidf_i = tf_i \cdot idf_i$$

n_i : キーワード i の出現回数

$|D|$: 総カテゴリ数

$|\{d : t_i \in d\}|$: キーワード i を含むカテゴリ数

ている。そのため、 tf と idf をかけた値である $tfidf$ を重みとして設定することで、カテゴリを特徴づけるキーワードの重みを適切に設定することができる。

2.3.3 上位カテゴリの重みの自動計算

上位カテゴリの重みの自動計算を行うために、カテゴリの上下関係を利用する。上位カテゴリを詳細化したものが下位カテゴリなので、その上位カテゴリと下位カテゴリの関連度を以下のように定義する。

【定義】(カテゴリの上下間の関連度)

$c, c' \in C$, cEc' が成り立つとき、 $R(c, c')$ は c, c' 間の関連度を表す値であり、 $0 \leq R(c, c') \leq 1$ が成り立つとする。□

下位カテゴリが存在する全ての上位カテゴリとその下位カテゴリに、関連度 R を 0 から 1 の範囲で設定する。設定した関連度と既に設定されている下位カテゴリの重みを用いて、以下の計算方法により上位カテゴリの重みを決定する。

計算に利用するキーワードは 2.2 節で構築したキーワード群である。 gf_i はカテゴリの文章の文章量の差をなくすため正規化しており、 $0 \leq gf_i \leq 1$ の関係が成り立つ。 gf_i のキーワードの出現頻度だけを用いて重みを計算すると、頻繁に登場する一般的なキーワードの重みが大きくなってしまふ。そこで idf_i は一般的なキーワードに対してのフィルタとして働き、多くのカテゴリの文章に出現するキーワードは重みが下がり、特定のカテゴリの文章にしか出現しないキーワードの重みを上げる役割を果たし

【上位カテゴリの重みの計算方法】

$c, c_1, c_2, c_3, \dots, c_n \in C, cEc_1, cEc_2, cEc_3, \dots, cEc_n$ が成り立つとき、上位カテゴリ c の重み $W(c, k)$ は

$$W(c, k) = \max(W(c_1, k) \cdot R(c, c_1), W(c_2, k) \cdot R(c, c_2), \\ W(c_3, k) \cdot R(c, c_3), \dots, W(c_n, k) \cdot R(c, c_n))$$

で表される。この時、 $\max(x_1, x_2, x_3, \dots, x_n)$ は括弧内の配列要素のうち、最大値を選択する関数である。

2.4 カテゴリへのマッピング

2.4.1 カテゴリへのマッピング方法

ここでは、2.1 節～2.3 節のプロセスで得られたカテゴリ集合とキーワード集合、重みを用いて、Web ページをカテゴリへマッピングする方法について述べる。

まず、対象となる Web ページの文章中に含まれるキーワード集合の全てのキーワードの出現回数を求める。以下に Web ページとキーワードに関する定義を示す。

【定義】 (Web ページ集合, Web ページのキーワードの出現回数, Web ページの特徴ベクトル)

- (1) $P = \{p_1, p_2, p_3, \dots, p_n\}$ を Web ページ集合とする。
- (2) $p \in P, k \in K$ のとき、 $O(p, k)$ は Web ページ p に含まれるキーワード k の出現回数を表す。
- (3) $p \in P, k_1, k_2, k_3, \dots, k_n \in K$ のとき、 $PV(p) = (O(p, k_1), O(p, k_2), O(p, k_3), \dots, O(p, k_n))$ は、Web ページ p に対する全てのキーワードの重みを表したものであり、 $PV(p)$ を Web ページ p の特徴ベクトルとする。

Web ページ p のキーワードの出現回数を求めた後、得られた Web ページ p の特徴ベクトル $PV(p)$ と全てのカテゴリの特徴ベクトル $CV(c)$ の内積和を求める。式で表すと以下ようになる。

【カテゴリの特徴ベクトルと Web ページの特徴ベクトルの内積和】

$$PV(p) \times CV(c) = \sum_{i=m}^n O(p, k_m) \cdot W(c, k_m)$$

求めた内積和が最も高い値を示したカテゴリに Web ページはマッピングされる。

2.4.2 山登り法を用いたマッピング

定義するカテゴリやキーワード、重みは不変ではないものの、変更の機会はいくつ多くはない。実際に Web ページをカテゴリにマッピングするアプリケーションを

考えた場合に、必要な計算は 2.4.1 節で述べた Web ページをカテゴリにマッピングする部分である。レスポンスタイムを考慮すると、カテゴリやキーワード数が増大したとき、計算量も増加傾向にある。すると、レスポンスタイムがかなりすぎてしまい、ユーザに対するストレスになる。そこで、全てのカテゴリに対して Web ページとの内積和を求めるのではなく、木構造であるカテゴリの構造を活かし、山登り法を用いたマッピングを考える。

山登り法とは、現在の解の近傍の内でも最も評価の高い解を近傍解として選び、現在の解より近傍解の方が高い評価の場合に近傍解と現在の解を入れ換える局所探索法である。

本手法に山登り法を採用すると、Web ページをカテゴリにマッピングする際に採用できる。計算量を飛躍的に抑えられるため、レスポンスタイムを小さくすることが可能である。山登り法を採用したカテゴリへのマッピング手法の手順をフローチャートで表すと図 1 のようになる。

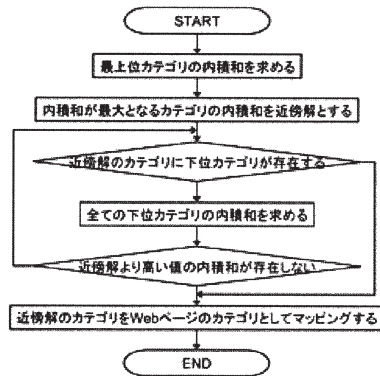


図 1. 山登り法を用いたマッピングのフローチャート

山登り法には、最適化問題と呼ばれる問題がある。最適化問題とは、全ての解を探索するわけではないので、いわゆる最大値である最適解ではなく、局所最適解が選択される可能性があるという問題である。この問題は本手法に採用した場合にも起こりうる。本手法の場合は、2.2.3 節で述べたように、上位のカテゴリの重みは下位のカテゴリの重みが反映されている。そのため、山登り法のアルゴリズムを採用した場合に最適解が得られる可能性が高いと考えられる。

3 応用事例

3.1 意味に基づいたサイトマップの自動構築

3.1.1 Web サイトのサイトマップ

一般に公開されている Web サイトには、サイトマップを用意している Web サイトも多く存在する。サイトマップは Web サイトの管理者が Web ページの意味を理解した上で、Web サイトに含まれる Web ページを階層的に表現し、どのようなページが存在するのかが分かるようにするものである。しかし、Web サイトによってはサイトマップを公開していない Web サイトもあり、その場合はどこにどのような Web ページが存在するのかが把握しづらい。そこで、Web サイトのリンク構造やディレクトリ構造ではなく、Web ページの意味に基づいたサイトマップの自動構築を本研究の提案手法を用いることで実現する。

3.1.2 Web サイト内の Web ページの抽出

サイトマップを作るために、Web サイト内に含まれる Web ページを抽出する必要がある。一般に Web サイトは 1 つのドメインの上で成り立っている。Web はハイパーリンクによって、Web ページ同士をつなげることができるので、ハイパーリンクのリンク先が同一ドメインである Web ページを逐次抽出していけば、Web サイトの Web ページを抽出することができる。具体的には、HTML の `<a>` タグに含まれる `href` がリンク先を表す属性なので、属性値が同一ドメインの URL である Web ページを抽出していけば良いことになる。

3.1.3 サイトマップの自動構築

抽出された Web ページに対して、本研究の提案手法である Web ページのカテゴリへのマッピング処理を行う。マッピング処理のイメージを図 3 に示す。マッピングされた Web サイトは Web ページに含まれる意味も考慮されて分類される。分類されたカテゴリは木構造になっているので、そのままカテゴリに従って Web ページを出力すれば、意味に基づいたサイトマップが完成する。

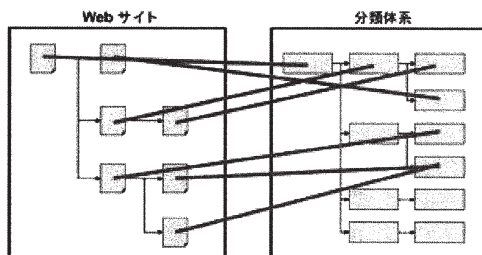


図 3. Web サイトのマッピングイメージ

3.2 意味に基づいた Web ページ検索

3.2.1 シソーラス検索

これまで、キーワードの重みをカテゴリの特徴ベクトル

ルまたは Web ページの特徴ベクトルとして扱っており、カテゴリまたは Web ページの観点からでのみ利用してきた。ここで、キーワードの観点で重みの値を利用することを考えてみる。

例えば、表 2 のような重みを与えられたとする。ここ

表 2. キーワードの重み付けの例

	カテ ゴリ 1	カテ ゴリ 2	カテ ゴリ 3	カテ ゴリ 4	カテ ゴリ 5	カテ ゴリ 6	カテ ゴリ 7
キー ワード 1	0.0	0.4	0.3	0.2	0.0	0.2	0.9
キー ワード 2	0.2	0.8	0.0	0.0	0.0	0.0	0.0
キー ワード 3	0.2	0.0	0.8	0.5	0.0	0.4	0.0
キー ワード 4	0.2	0.0	0.9	0.5	0.0	0.3	0.0
キー ワード 5	0.0	0.0	0.0	0.0	1.0	0.1	0.0

で着目すべきはキーワード 3 とキーワード 4 である。これらのキーワードの重みは、0 以外の値が現れるカテゴリが同じであり、カテゴリ 3 の重み (0.8 と 0.9) とカテゴリ 4 の重み (0.4 と 0.3) に 0.1 ずつの違いが現れる程度である。このことから、キーワード 3 とキーワード 4 が全てのカテゴリに対して与える意味がほぼ同じであると言える。すなわち、キーワード 3 とキーワード 4 は同義語または類義語 (シソーラス) であるとも言える。これをキーワード検索に応用し、キーワード 3 で検索を行ったときに、同時にキーワード 4 の検索結果も反映させることで、ユーザが要求している情報を持つ Web サイトが検索結果として現れやすくすることができる。

3.2.2 検索結果のランキングアルゴリズムへの応用

これまでは Web ページとカテゴリとのマッピング情報は蓄積せずに、リアルタイムでのレスポンスを考えていたが、マッピング情報をサーバに蓄積することでも活用の可能性は十分にある。

ロボット型検索エンジンの一般的な検索方法は、キーワード検索である。入力したキーワードを含む Web ページを検索エンジン毎に異なるランキングアルゴリズムに従って検索結果を表示する。しかし、これではキーワードは単なる文字列の集合としてしか扱われておらず、キーワード自体には何の付加情報も与えられていない状態である。そこで、本研究で提案した Web ページとカテゴリとのマッピング情報を利用することで、キーワードに意味を持たせ、ランキングアルゴリズムの 1 つとしての応用事例を提案する。

まず、検索対象となる全ての Web ページをカテゴリにマッピングする。次に、本手法で抽出したキーワード群

のキーワードをカテゴリにマッピングさせ、キーワードに付加情報を与える。キーワードをマッピングするカテゴリは以下の方法で選択する。

ユーザがロボット型検索エンジンを用いて、キーワード検索を行ったときに、キーワード群に含まれるキーワードを検索した場合を考える。キーワードはカテゴリにマッピングされているので、その情報からユーザが検索したいカテゴリを把握することができる。Web ページとカテゴリのマッピング情報を活かして、検索したキーワードにマッチした Web ページのうち、検索したいカテゴリにマッピングされている Web ページを検索結果の上位として表示することが可能となる。また、Web ページとカテゴリのマッピングはカテゴリの特徴ベクトルと Web ページの特徴ベクトルの内積を用いて計算されているので、内積の値をランキング決定のための値として用いることも考えられる。

4 実験と考察

4.1 サイトマップ自動構築システム

4.1.1 システムの概要

本システムは Web ブラウザを通して利用する Web アプリケーションの形をとる。ユーザは用意されたフォームにサイトマップを構築したい Web サイトの URL を入力する。すると、自動的に意味に基づいて構造化されたサイトマップを表示し、各 Web ページへのハイパーリンクを生成するシステムである。システムのスクリーンショットを図 4 に示す。



図4. システムのスクリーンショット

Web ページ数は Web サイトの規模によって異なる。Web ページが多い Web サイトの場合は、全ての Web ページをマッピングした後にサイトマップの構築をすると非常に時間がかかってしまう。この問題を解決するため、順次カテゴリに分類された Web ページからサイトマップに反映する仕組みにした。それを実現するための技術

として、Ajax(Asynchronous JavaScript + XML)を用いている。Ajax はサーバのレスポンスを Web ページの遷移なしで、表示されている Web ページに反映することができる非同期通信を実現する技術である。

4.1.2 カテゴリの構築

本システムで用いるカテゴリは、ディレクトリ型検索エンジンでもあるYahoo!のYahoo!カテゴリ^[2]を参考に構築した。本実験では、Yahoo!カテゴリの最上位カテゴリである“コンピュータとインターネット”の下位カテゴリにあたる“ソフトウェア”を対象を絞ってカテゴリを構築した。Yahoo!カテゴリの画面のスクリーンショットを図5に示す。一部不要なカテゴリについては削除した。例えば、“ソフトウェア”というカテゴリの下位カテゴリにあたる“団体”や“評論・レビュー”などである。“団体”や“評論・レビュー”は他の“ソフトウェア”の下位カテゴリとの関係とは異なり、上位カテゴリを詳細化するものではないので除外している。本実験でのカテゴリ数は98になった。

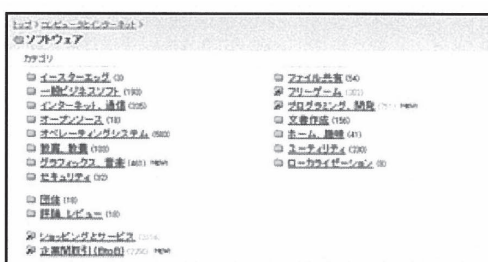


図5. Yahoo!カテゴリのスクリーンショット*

4.1.3 キーワードと重みの設定

本実験でのキーワードと重みは手動で設定した。Yahoo!カテゴリに登録されている Web サイトを閲覧し、カテゴリに関連するキーワードを人的に抽出し、362 のキーワードを得た。Yahoo!カテゴリ重みはカテゴリに関連するキーワードでも一般語は重みを低く設定するようにし、tf-idfについても考慮した。最下位のカテゴリの重みのみを設定し、上位カテゴリの重みは2.3.3 節で述べたアルゴリズムを用いて計算している。上下間のカテゴリの関連度の強さを4段階に分け、関連度の強い順に関連度 $R(c, c')$ を0.8 0.7 0.6 0.5 の値を設定している。

4.2 実験の考察

4.1 節で構築したサイトマップ自動構築システムを用いて、本提案手法の検証を行った。

*¹ 出典：Yahoo! カテゴリ

図4に示したシステムのスクリーンショットはYahoo!カテゴリ内のソフトウェアのカテゴリを表示したページのURLを入力して、サイトマップ構築を行った図である。本システムで構築したカテゴリや抽出したキーワードはYahoo!カテゴリを参考にしているため、Yahoo!カテゴリ内のページの約6割は適切なカテゴリにマッピングされていた。また、マッピングされるべきカテゴリの上位カテゴリまたは下位カテゴリにマッピングされていたページが残り約4割を占めており、Yahoo!カテゴリ内のページにおいては全く関係ないカテゴリにマッピングされるページはほとんどない結果となった。

しかし、通常のWebサイトを本システムに入力すると、それほど高い精度は得られなかった。今回の実験では”ソフトウェア”以下のカテゴリを対象にしているため、それらのカテゴリにマッピングされるべきWebサイトのURLを本システムに入力した。マッピングされた1つ1つのWebページを見ていくと、本来マッピングされるべきカテゴリにマッピングされているWebページは少なく、2,3割程度のWebページしか適切なカテゴリにマッピングされておらず、最上位のカテゴリである”ソフトウェア”にマッピングされるWebページが多くなってしまった。このように、精度が低くなってしまった原因はいくつか考えられる。

前述した最上位カテゴリにマッピングされてしまうという問題は、上下関係の関連度を高く設定しすぎたため起こったと考えられる。本実験では関連度を0.8~0.5の範囲で設定していたが、もう少し低く設定した方がよいのかもしれない。

2つ目の原因としては、広告やスポンサー情報などの本来Webページが提供したい情報とは関係ない情報が含まれていることである。この問題に関しては、Webページをカテゴリにマッピングする際に、Webページが本来提供したい部分だけをマッピングのための計算に利用する仕組みが必要になる。

3つ目は、Webページの情報量不足の問題である。Webページ自体に文章が少ないWebページ、FlashなどのメディアコンテンツがメインになっているWebページ等はマッピングするための情報が少なく、精度が低くなってしまふ。この問題に関しては、システム側では解決することができない問題である。

最後に、キーワード数の不足や重みの妥当性の問題もある。本実験では、キーワードの抽出や重みの設定は全て手動で行ってきた。マッピングに必要なキーワードとして充足しているか、重みの設定は妥当であったかについてはまだまだ検討の余地が残る。

5 おわりに

本研究の提案手法では、Webページとカテゴリを完全に1対1でマッピングしていた。しかし、カテゴリの分類の仕方によっては、Webページの内容が必ずしも1つのカテゴリにマッピングされるとは限らない。Webページとカテゴリが1対多の関係を持たせるような拡張可能性も十分に考えられる。

カテゴリの階層関係について言えば、本提案手法では上位カテゴリと下位カテゴリは詳細化の関係、いわゆるis-a関係のみに限定していた。カテゴリ構築の際に参考にしたYahoo!カテゴリでは、is-a関係以外の関係も含まれている。Webページを適切に分類するためのカテゴリ構造の構築方法についても検討すべき課題として挙げられる。

本研究の一番の課題はマッピングの精度である。本研究でWebページの自動分類のための体系を構築することができた。しかし、マッピングの精度は決して高いと言えるものではなかった。第3章で紹介した応用事例のように、実際のアプリケーションとして機能させるためには、もっと高い精度が要求される。マッピングの精度を向上させることが本研究の最大の課題となるだろう。

参考文献

- [1] Netcraft. <http://news.netcraft.com/>.
- [2] Yahoo!カテゴリ. <http://dir.yahoo.co.jp/>.
- [3] 栗原聡, 廣津登志夫, 高田敏弘, 明石修, 菅原俊治: リンク情報におけるWebページ間の類似度推定, コンピュータソフトウェア, Vol.18, No.6(2001), pp.15-26
- [4] 池辺正典, 田中成典, 吉田均, 中村健二, 小林健太: Webリンク構造解析と自然言語処理による組織関係の抽出についての研究, 情報処理学会論文誌, Vol.47, No.6(2006), pp.1687-1695
- [5] 安形輝, 石田栄美, 久野高志, 野末道子, 上田修一: WWWページの自動分類NDCの分類体系とYahooのカテゴリを使った分類, 情報処理学会研究報告(99-FI-54), Vol.99, No.39(1999), p.113-120
- [6] 石田栄美, 久野高志, 安形輝, 野末道子, 上田修一: 内容的なまとまりをもつWebページ群の自動判定, 1999年度三田図書館・情報学会研究大会発表論文集, 1999, p.21-24
- [7] 落谷亮: WWWページの分類におけるテキストの特徴分析手法, 情報処理学会研究報告・自然言語処理研究会報告, Vol.97, No.29(1997), pp. 85-90
- [8] 大西高裕, 山名早人: リンク構造を用いたWebページ自動分類の精度向上法, 情報処理学会第65年全国大会, 2003